

may not be the current version because the most recent revision to a content container of the shared object is not available from the store.

[0075] Advancing to block 820, a request tag and client information are assigned to the store to indicate that the client requires the most recent revision data to update the shared object. The client information may include a GUID that identifies the requesting client and a time stamp that identifies the time when the client requested the current version of the shared object from the store.

[0076] Transitioning to block 830, the current version of the shared object is received at the store. The store may receive the current version of the shared object when another client accesses the store with the most recent revision data. The requesting client is informed that the current version of the shared object has been received by the store. Continuing to block 840, the current version of the shared object is synchronized with the requesting client. Processing then terminates at the end block.

[0077] FIG. 9 illustrates an operational flow diagram illustrating a process for seamlessly transitioning from asynchronous to synchronous communication modes. The process begins at a start block where a peer group is established that identifies users who are authorized to access a shared object.

[0078] Moving to block 900, a client accesses the shared object on a server. The client is automatically connected to other clients that are also accessing the shared object (i.e., the peer group). The shared object is associated with a manifest file. The shared object includes a unique location identifier that identifies the location where the corresponding manifest file is stored in the system.

[0079] Proceeding to block 910, the manifest file is retrieved from the location identified by the unique location identifier. The manifest file identifies the locations where other versions and instances of the shared object are stored within the system. The manifest file includes a peer group identifier for the peer group where a version of the shared object is stored.

[0080] Advancing to block 920, a peer-to-peer network is established when any other client in the peer group accesses a version or instance of the shared object identified by the manifest file. Thus, the client may disconnect from the server and continue to access the shared file on the peer-to-peer network. Processing then terminates at an end block.

[0081] FIG. 10 illustrates an operational flow diagram illustrating a process for seamlessly transitioning from synchronous to asynchronous communication modes. The process begins at a start block where a peer-to-peer network is established between at least two users who are authorized to access a shared object.

[0082] Moving to block 1000, a client accesses the shared object on the peer-to-peer network. The shared object is associated with a manifest file. The shared object includes a unique location identifier that identifies the location where the corresponding manifest file is stored in the system.

[0083] Proceeding to block 1010, the manifest file associated with the shared object is retrieved from the location identified by the unique location identifier. The manifest file identifies the locations where other versions and instances of

the shared object are stored within the system. Advancing to block 1020, the client connects to a server. The client determines which other clients are also connected to the server. Transitioning to block 1030, the client identifies other clients that are authorized to access the shared object from the peer-to-peer network. Continuing to block 1040, the client connects to an authorized client when the peer-to-peer network is unavailable. Processing then terminates at an end block.

[0084] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

What is claimed is:

1. A computer-implemented method for synchronizing multiple user revisions to a shared object, comprising:

receiving a revision to the shared object;

determining whether the revision conflicts with a synchronized revision on a master page of the shared object; and

synchronizing the revision with the shared object when the revision is determined to be associated with a current version of the shared object and when the revision is determined to be not conflicting with a synchronized revision.

2. The computer-implemented method of claim 1, further comprising:

displaying the revision on a conflict page associated with the master page when the revision is determined to be conflicting with a synchronized revision;

displaying a conflict indicator on the master page, wherein the conflict indicator indicates that the master page is associated with the conflict page; and

merging the revision into the shared object.

3. The computer-implemented method of claim 2, wherein displaying the revision further comprises:

displaying the revision on the conflict page in a location corresponding to the location where the synchronized revision is displayed on the master page; and

highlighting the revision on the conflict page.

4. The computer-implemented method of claim 2, further comprising displaying the conflict page and the associated master page when the conflict indicator is selected.

5. The computer-implemented method of claim 1, further comprising:

determining whether the revision is irrelevant; and

purging the revision that is determined to be irrelevant.

6. The computer-implemented method of claim 1, further comprising:

receiving a client request for a current version of the shared object at a store;

receiving the current version of the shared object at the store; and

synchronizing the current version of the shared object with the client.